



# **MUKAUTUVA VERKKOSUUNNITTELU**

Antti Tolppanen

Opinnäytetyö  
Toukokuu 2011  
Viestinnän koulutusohjelma  
Visuaalinen suunnittelu

TAMPEREEN AMMATTIKORKEAKOULU  
Tampere University of Applied Sciences

# OPINNÄYTTEEN TIIVISTELMÄ

**Antti Tolppanen**

***Mukautuva verkkosuunnittelu***

Toukokuu 2011

45 sivua

Tampereen ammattikorkeakoulu

Viestinnän koulutusohjelma

Visuaalinen suunnittelu

Lopputyön muoto: kirjallinen

Lopputyön ohjaaja: Ari Närhi

Joustavarakenteiset verkkosivut, eli päätelaitteen näytön tai selainikkunan koon mukaan asettuvat CSS-rakenteet voivat olla tyylimäärityksiltään taloudellisia vähentäessään monien eri päätelaittekohtaisten CSS-tyylimääritysten tarvetta. Ohjeet eri elementtien CSS-tyylimäärityksille löytyvät eri lähteistä. Näiden ohjeiden kokoaminen yhteen antaa keinot toteuttaa verkkosivuilla tehokkaamman päätelaitteille ja selainikkunaan mukautuvan CSS-määrittelyn.

Huomioimalla suunnittelussa verkkosivujen sisällössä käytetyimpien elementtien asettelu ja niiden suhteet ympäröiviin rakenteisiin pystytään vaikuttamaan siihen miten eri päätelaitteet piirtävät sisältöä näytöille. Lisäksi suunnittelussa on hyvä huomioida kosketusnäyttöjen vaatimuksia.

Todennäköisintä on että CSS-määrittelyssä joudutaan käyttämään pikseli-, em- ja prosenttiperustaista määrittelyä toimivimman rakenteen saavuttamiseksi. Rakenteellisesti toimivimmat verkkosivut, joita on miellyttävä käyttää sekä kosketusnäytöillä, että hiiriohjauksella, ovat pystysuuntaisia ja ne hyödyntävät JavaScript- ja jQuery-liitännäisiä.

---

Avainsanat: internet käytettävyys

# THESIS SUMMARY

**Antti Tolppanen**

***Responsive Web Design***

May 2011

45 pages

TAMK University of Applied Sciences

Media Programme

Area of specialisation: Art of Visual Design

Type of Final Project: Written

Thesis supervisor: Ari Närhi

## **Abstract:**

Web sites with liquid layout structures can lessen the need of specific CSS-definitions for multiple terminals. Liquid layout means CSS-structures which can observe the size of browser window or terminals monitor. The notes for sizing and giving proper values for different elements used in web design can be found from many different sources. By collection these notes one can improve CSS-definitions and in result write better CSS

Putting weight on the most basic elements of the web pages and defining properly the distances and margins of these elements it is possible to determine how different kind of terminals render the content to the screen. As a designer It is also important to consider the requirements of touch screen devices.

It is recommended that the CSS-definitions use a pixel, em, and the percentage-based definition in the structure to achieve the best structure. Structurally functional websites which are comfortable to use with touch screens or mouse are built vertically and they use JavaScript- and jQuery-plugins.

---

Keywords: internet usability

# Sisällys

<b>1</b>	<b>Johdanto.....</b>	<b>3</b>
<b>2</b>	<b>Yleisiä termejä verkkosuunnittelussa.....</b>	<b>5</b>
2.1	Graafinen käyttöliittymä.....	5
2.2	.htaccess.....	5
2.3	Käyttöliittymä.....	5
2.4	JavaScript .....	5
2.5	Julkaisujärjestelmä, eli CMS .....	6
2.6	jQuery .....	6
2.7	PHP.....	6
2.8	Käytettävyys.....	6
<b>3</b>	<b>Verkkosivun rakenteet .....</b>	<b>8</b>
3.1	Sisällön ja navigoinnin suhde.....	11
3.2	Navigointi .....	13
3.3	Lomakkeet .....	14
<b>4</b>	<b>Kosketusnäyttöjen erityispiirteitä .....</b>	<b>17</b>
4.1	Aktiivinen kosketusalue.....	18
4.2	Käyttäjän oikea- ja vasenkätisyys .....	19
4.3	Navigoinnin rakenne ja asettelu kosketusalustoilla .....	19
<b>5</b>	<b>Verkkosivusuunnittelu kosketusnäytöille .....</b>	<b>21</b>
5.1	Verkkosivun valikkorakenteet kosketusnäytöillä .....	21
5.2	iDiD.fi verkkosivusto ja kosketusalustakäyttö.....	23
<b>6</b>	<b>Mukautuvan verkkosuunnittelun keinot .....</b>	<b>26</b>
6.1	Päätelaitetunnistus ja käytettävän CSS-tyylin määrittäminen .....	27
6.2	Päätelaitteen näyttöön mukautuva CSS-määrittely .....	27
6.3	CSS-tyyliin nollaus.....	28
6.4	Mukautuvat rakenteet.....	30

6.5	Tekstin CSS-määrittely .....	31
6.6	Kuvien koon CSS-määrittely .....	32
<b>7</b>	<b>Mukautuva verkkosuunnittelu käytännössä .....</b>	<b>34</b>
<b>8</b>	<b>iDiD-verkkosivuston kehitysehdotuksia.....</b>	<b>36</b>
8.1	Navigointi .....	36
8.2	Linkit.....	37
8.3	Pystysuuntaan jäsennetty verkkosivu .....	38
<b>9</b>	<b>Yhteenvetoa .....</b>	<b>40</b>
	<b>Lähteet .....</b>	<b>43</b>

# 1 Johdanto

Käsittelen opinnäytetyössäni mukautuvaa verkkosuunnittelua. Mukautuvalla verkkosuunnittelulla pyritään huomioimaan internetin selauksessa käytettävien laitteiden ja selainten erilaisia ominaisuuksia ja rajoituksia siten, että pystytään mahdollisimman tehokkaasti tekemään useilla eri alustoilla yhdenmukaisesti toimivia verkkosivuja. Opinnäytetyön aiheen tarkoituksena on vähentää eri alustakohtaisten tyylimäärittelyjen tarvetta ja tehostaa suunnittelutyötä ja CSS-määrittystä.

Ohjeita, artikkeleita ja oppaita mukautuvasta verkkosuunnittelusta löytyy verkosta ja kirjallisuudesta eri lähteistä. Opinnäytetyössäni olen halunnut kasata niitä yhteen. Tämä on tehostanut omaa työskentelyäni verkkosivujen rakenteita suunnitellessani ja vähentänyt tarvetta palata samojen verkkosuunnitteluoppaiden pariin ongelmia kohdatessani. Koottu ja tiivistetty lähdemateriaali toimii siis oppaana verkkosivustojen perusrakenteita suunniteltaessa.

Haasteena eri alustoille suunniteltaessa ovat erilaiset näyttöjen resoluutiot ja näytön pysty- tai vaakasuuntaisuus. Ratkaisuja näihin haasteisiin ovat joustavat rakenteet sisällön esittämisessä, eli tekstisisältöjen ja kuvien asettuminen sisällössä siten ettei niille määritellä kiinteitä leveyksiä, vaan suurimpia ja pienimpiä sallittuja arvoja verkkosivun selaamiseen käytettävän päätelaitteen näytön, tai verkkoselaimen ikkunassa näkyvän alueen prosenttiosuutena.

Mukautuvan verkkosuunnittelun lisäksi huomioin opinnäytetyössäni kosketusnäyttöjen asettamia vaatimuksia käytettävyydelle verkkosivuja suunniteltaessa. Tällaisia huomioitavia erityispiirteitä löytyy käyttöliittymärakenteita suunniteltaessa navigoinnin toiminnallisten elementtien koossa ja sijoittelussa.

Yhdistämällä verkkosuunnittelussa tehokkaan CSS-tyylisuunnittelun ja kosketusnäyttöisten verkkosivujen selaukseen käytettävien alustojen käytettävyyseikat, voi vähentää eri tyyli tiedostojen tarvetta ja myös tehdä verkkosivuista pitkäikäisempiä ja käytettävämpiä. Koska verkkosivuja selataan sormitietokoneilla, älypuhelimilla ja tietokoneilla käyttäen navigointiin sekä

kosketusnäyttöjä että hiirtä ja joskus molempia on perusteltua tehostaa suunnittelua sen sijaan, että sivun tyylit määritellään päätelaitekohtaisesti.

Käsittelen opinnäytetyössäni pääpiirteitä verkkosivuston toiminnallisten elementtien asettelusta, käytettävyydestä ja luettavuudesta ja yhdistelen niitä lähteissä määriteltyihin kosketusnäyttöjen käytettävyyseikkoihin.

Käsitellessäni tehokasta tyylisuunnittelua eri päätelaitteille käytän termiä mukautuva verkkosuunnittelu. Englanninkielisissä lähteissä aihetta käsitellään usein otsikolla Responsive Web Design.

Esittelen työssäni Connecting Talents Oy:lla vuosien 2010 ja 2011 vaihteessa koostamaani verkkosivua jonka ulkoasusuunnittelussa olen pohtinut käytettävyyttä kosketusalustoilla HTML-pohjaisessa verkkosivustossa. Verkkosivu löytyy osoitteesta <http://www.idid.fi>. Tarkastelen sivustoa myös mukautuvan verkkosuunnittelun kannalta ja pohdin sopivia kehitysehdotuksia.

Esittelen muiden esimerkkien yhteydessä vaihtoehtoja eri päätelaitteille sopivimmaksi kokemaani verkkosivurakennetta esimerkin kautta, jossa on pyrin hyödyntämään opinnäytetyössäni kokoamaa materiaalia. Esittelemäni malli on työversio iDiD-verkkosivustoa seuranneesta projektista. Siinä on hyödynnetty jQueryä, JavaScriptiä ja mukautuvia rakenteita. Mallisivu on upotettu MODX-julkaisujärjestelmään.

## **2 Yleisiä termejä verkkosuunnittelussa**

### **2.1 Graafinen käyttöliittymä**

Graafinen käyttöliittymä, tai GUI (Graphical User Interface) on tekstistä, kuvista ja muista käyttöliittymäelementeistä muodostuva tapa ohjata sovellusta (Graafinen käyttöliittymä 2010). Graafisen käyttöliittymän on hyvä olla jo alkuvaiheessa osana suunnitteluprosessia toimivan kokonaisuuden saavuttamiseksi, sillä toimiva graafinen ilme on myös osa käytettävyyttä ja kilpailutekijä (Kalimo 1996, 141-142).

### **2.2 .htaccess**

Yleisen Apache-HTTP-palvelinohjelman pääasetuksien sääntöjä voidaan ylikirjoittaa .htaccess-oletustiedostolla, tätä voidaan käyttää uudelleenohjaukseen, asettaa oletusvirheilmoitus sivu ja määrittää salasana-asetuksia (What is .htaccess? 2010).

### **2.3 Käyttöliittymä**

Käyttöliittymä on se osa ohjelmaa tai verkkosovellusta, jolla sitä hallitaan ja käytetään. Tämä on myös käyttäjälle näkyvä osa. Tähän kuuluvat valikkorakenteet, napit ja visuaaliset palautteet toiminnallisuuksista, joskus myös äänipalaute. Käyttöliittymän on tarkoitus myös ohjata käyttäjää ja sillä käyttäjä navigoi ja on vuorovaikutuksessa sovelluksen kanssa. (Kalimo 1996, 7.)

### **2.4 JavaScript**

JavaScript on Javasta yksinkertaistettu skriptikieli. Skriptikieliä käytetään erityisesti lisätoimintojen ja sovelluksien laajentamiseen eripituisina koodiriveinä hyvinkin lyhyistä toiminnoista kokonaiseen sovellukseen asti. JavaScriptillä nämä toiminnot voidaan rajata ulottumaan koko dokumenttiin tai vain tiettyyn osaan sitä. (Peltomäki 2000, 6-11.)

JavaScript sopii muokattavuutensa puolesta hyvin HTML-dokumentteihin, koska siihen tehtävät muutokset näkyvät helposti selaimen näkymä päivitettäessä. Selaimet tulkitsevat JavaScriptiä JavaScript-tulkilla. (Peltomäki 2000, 6-11.) JavaScript on



internetin käytetyimpiä kieliä. Se on yhteensopiva suosituimpien selainten kanssa ja sitä käytetään lisäämään sivujen interaktiivisuutta (JavaScript Introduction 2011). JavaScriptillä voidaan myös tunnistaa käyttäjän selain tai päätelaite (Mike 2010).

## **2.5 Julkaisujärjestelmä, eli CMS**

Suurin osa verkkosivuista julkaistaan rakennettuna jonkin julkaisujärjestelmän päälle toimimaan dynaamisesti. Julkaisujärjestelmät helpottavat WYSIWYG-editorilla loppukäyttäjän sisällönhallintaa. Julkaisujärjestelmä-termi viittaa nimensä mukaisesti julkaisemiseen. Sen englanninkielinen nimi on CMS (Content Management System).

Sisällönhallintajärjestelmällä viitataan usein laajempiin järjestelmiin, vaikka näitäkin käytetään joskus ristiin asiayhteydestä riippuen. Sisällönhallintajärjestelmällä tarkoitetaan kuitenkin nimen mukaisesti sisällönhallintaa, ei julkaisemista. (Tolvanen, 2009.)

Käytän opinnäytetyössäni termiä julkaisujärjestelmä.

## **2.6 jQuery**

jQuery on selainyhteensopiva open source -JavaScript-kirjasto. Se on tehty tehostamaan verkkosivujen skriptipohjaisten dynaamisten sisältöjen koostamista (jQuery 2010).

## **2.7 PHP**

PHP on ohjelmointikieli jota käytetään dynaamisilla internet-sivuilla. Se kommunikoi palvelimen tietokantojen kanssa ja sitä voidaan myös käyttää selaimen (PHP Introduction 2011), tai päätelaitteen tunnistamiseen (Mike 2010).

## **2.8 Käytettävyys**

Sovelluksen käytettävyyttä voi tarkastella sillä, miten hyvin se täyttää tehtävänsä. Käytävyyden mittaaminen taas on lähes mahdotonta käsitteen sisältämien tarkoituksien vuoksi, vaikka sitä määritteleviä standardeja onkin määritelty. Käytettävyys myös koetaan aina subjektiivisesti. Käytettävyyden takia onkin hyvä pitää käyttöliittymä

informaatioltaan mahdollisimman suppeana, mutta selkeänä. Tähän kannattaa panostaa, koska hyvä käytettävyys on markkinoinnillisestikin järkevää. (Kalimo 1996, 8-13.)

Steve Krug (2006, 1-49) suosittelee kirjassaan *Älä pakota minua ajattelemaan*, että verkkosivuilla, ei pelkästään käyttöliittymiä suunniteltaessa, vaan myös sisällön suhteen yleisesti on teksti hyvä karsia niin vähiin kuin mahdollista käytettävyyden maksimoimiseksi ja kohinan minimoimiseksi. Kohinalla Krug (2006, 38) tarkoittaa web-sivuilla olevaa käyttäjän toimintaa häiritsevää epäolennaista informaatiota. Sisällön karsiminen myös nostaa oleellisen sisällön esille selkeämmin.

### 3 Verkkosivun rakenteet

Normaalisti verkkosivut rakentuvat seuraavasti: Header-osassa esitellään sivusto ja siihen sijoitetaan normaalisti myös navigointi. Sisältöalueella on kaikki sivuston oleellinen sisältö. Tämä sijoitetaan header-osan alle suunnilleen keskelle sivustoa. Sisältöalueen rinnalla, normaalisti vasemmalla puolella voi olla apunavigointi. Apunavigointi voi sijoittua myös oikealle ja sisältää usein blogien kaltaisilla sivustoissa pääsyn arkistoihin ja kategoriavalikoihin. Sivuston alalaitaan tulee footer-osa, jonne sijoitetaan normaalisti yhteystiedot, versio navigoinnista ja nykyään myös paljon muuta informaatiota kuten linkkejä muille sivuille ja yhteistyökumppaneiden logoja sekä yleistä informaatiota sivustosta tai ylläpitäjästä.

Tarkemmin määriteltynä Header-osaan, joka on web-sivuston yläreunaan muodostuva alue, sijoitetaan normaalisti sivuston tunnus tai logo. Päänavigointi suositellaan sijoitettavaksi yläreunaan ja vaakatasoon. (Krug 2006, 61.)

Sivuston lisäominaisuudet näkyvät header-osassa eriteltynä ja selvästi omina toimintoinaan (esimerkiksi haku-toiminto, kirjautuminen ja käyttöapu-valikko).



Kuva 1: Web-sovelluksen header-osan rakenne mukaillen Steve Krugin välilehtimallia.

Sisällön jaottelussa tyhjän tilan käyttö on tehokas tapa jäsentää ja erotella elementtejä toisistaan. Myös rajaavien alueiden muodoilla voidaan vaikuttaa siihen mihin osat yhdistyvät. Reunapyöritykset toimivat täsmällisiä neliöitä tehokkaampina visuaalisina

vihjeinä käyttäjälle. (Fadeyev 2009.) Esimerkiksi yhteydenottolomakkeiston rajaaminen pyöristetyllä taustalla tekee siitä kokonaisuuden.

Alatason navigoinnit tai paikalliset navigointivälineet sijoitetaan vasemmalle, normaalisti allekkain jäsennettynä ja tasot eroteltuina (Krug 2006 59–61).

Sivuston keskeinen sisältö esitetään normaalisti oikealla tai keskellä, riippuen palstajaosta. Sisältöalueelle ja sisällölle on varattava suurin tila. Sisältöosa on myös hyvä nimetä selkeästi noudattaen sinne ohjaavien linkkien nimiä. (Krug 2006 56–66, 71–73.)



*Kuva 2: Web-sovelluksen alatason navigoinnin sisältöosan rakenne mukaillen Steve Krugia*

Footer-osaan sijoitetaan normaalisti yhteystietoja, tai pieni versio navigoinnista se sijoitetaan sivuston alaosaan (Krug 2006, 59-61). Muita footeriin yleisesti sijoitettavia elementtejä ovat navigoinnin tai sivukartan ja yhteystietojen lisäksi linkit suositelluille sivuille. Footeriin voi sijoittaa myös linkin takaisin sivun alkuun ja pelkistetyn yhteydenottolomake. (Moline 2011).



*Kuva 3: iDiD.fi verkkosivun footer-osa. Footerin yläosassa linkit iDiD:n rinnakkaistuotteisiin ja Facebook-toiminto. Alaosassa vasemmalla versio päänavigoinnista. Oikealla iDiD.fi -palvelun tarjoavan Connecting Talents Oy:n yhteystiedot.*

Sivun alkuun palauttava toiminto voi olla pienemmällä vähään sisältöön perustuvilla sivuilla turhalta tuntuva toiminto, varsinkin jos koko näkymä asettuu näytölle kerralla. Sivun alkuun palauttavat toiminnot saattavat ankkureihin kohdistuessaan myös sotkea käyttäjän navigointia sivustolla: siirryttyään sivun alkuun esimerkiksi footeriin sijoitetulla palaa alkuun -toiminnolla käyttäjä palaa selaimen ”edellinen sivu”-toiminnolla footer-osaan, eikä edelliselle sivulle joka on ollut auki. (Leggett, Maier 2009, 132.)

Sivun visuaalisilla ratkaisuilla ja siirtymien animoinnilla voi ohjata käyttäjää kokemaan sivun osiot sivuiksi eikä yksittäisiksi kappaleiksi sivustolla.

Verkkosivun elementtien asettelu kokonaisuudessaan noudattelee seuraavaa rakennetta:



Kuva 4: Web-sovelluksen suositeltu sisältörakenne mukaillen Steve Krugia

### 3.1 Sisällön ja navigoinnin suhde

Verkkosivusuunnittelussa Steve Krug (2006, 7-39) painottaa selkeyttä ja intuitiivista, yleisesti totuttuihin tapoihin pohjautuvaa suunnittelua ja sijoittelua käyttöliittymäelementeille. Käytettävyyttä lisää erityisesti navigoinnin eri tasojen, kuten päätasen navigoinnin ja alavalikkojen sekä muiden linkkien selkeä erottaminen toisistaan.

Verkkosivun navigointi on olemassa käyttäjälle pysyvänä kiintopisteenä sivustolla. Navigointi on pidettävä pääosin muuttumattomana, sillä se toimii myös apuna käyttäjälle tämän hahmottaessa sijaintiaan sivustolla. Toimiva ja käytettävä kokonaisuus myös markkinoi sovelluksen tarjoajaa antaessaan kuvaa sivuston käytettävyydestä. (Krug 2006, 51-59). Tämän takia on järkevää noudattaa vakiintuneita käytäntöjä, joiksi voi lukea tiettyjen elementtien sijoittelun, kuten sivuston tunnuksen, navigoinnin ja sisältöalueen asemoinnin. Totuttujen tapojen noudattaminen auttaa käyttäjää nopeammin hahmottaan toimintojen sijainnit ja sivuston rakenteen.



Kuva 5: Web-sovelluksen elementtien asettelu Steve Krugia mukaillen

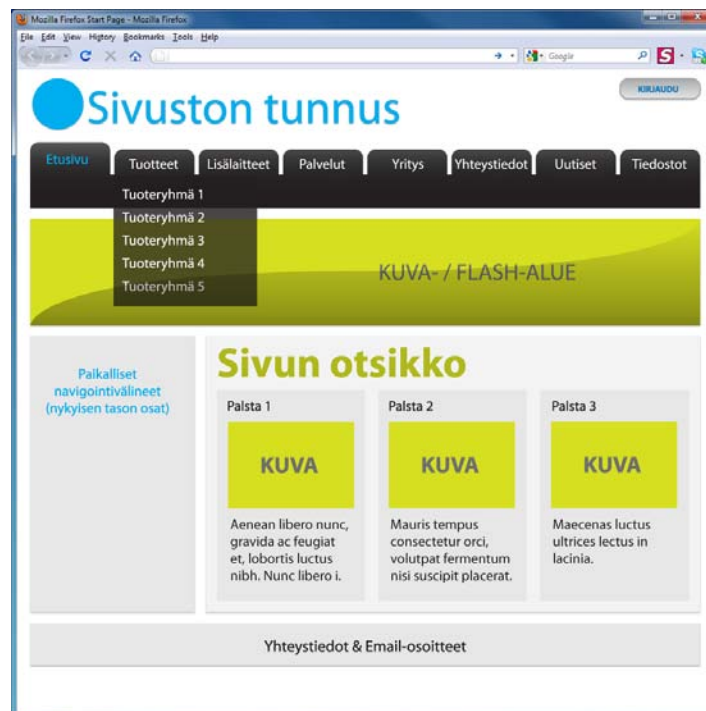
Vaikka perusrakenteita ei verkkosivustolla kannata juurikaan muuttaa siirryttäessä sivustolla sivulta toiselle, niin käyttöliittymäelementtien ja sisällön sijoittelua tulee joissakin tapauksissa jäsentää sen mukaan, millä alueella sivustoa liikutaan ja mitä halutaan käyttäjälle kertoa ja näyttää. Logiikan ja pääpiirteiden asettelussa on kuitenkin pysyttävä jatkuvasti käyttäjälle tunnistettavina ja helposti hahmotettavina. (Krug 2006, 78).

### **3.2 Navigointi**

Päänavigoinnissa Krug (2006, 79-84) suosittelee käytettäväksi “välilehteä”, eli navigointivalikkoa, joka nostaa linkin korostettuna esiin sen ollessa valittu tai aktiivinen. Valittua välilehteä on järkevää korostaa, jolloin se liitetään tehokkaammin viittaamaansa sisältöön. Tehokeinoina toimivat hyvin esimerkiksi rajaukset, värikorostukset ja kehystykset. Värikoodauksessa on Krugin mukaan kuitenkin omat ongelmansa käytössä, sillä merkittävä osa käyttäjistä ei huomioi sitä ollenkaan. Tästä huolimatta se on joskus toimiva tapa erotella sisältöä ja navigoinnin tasoja. Värikoodausta on voi käyttää lisätehosteena sen sijaan että se olisi ainoa tapa korostaa valintaa.

Välilehdet navigoinnissa ovat selkeytensä lisäksi erityisen toimivia kosketusnäytöillä, koska ne vaativat jo muotonsa puolesta ympärilleen tilaa. Välilehdet erottuvat hyvin toteutettuina muusta sisällöstä muodolla, värillä ja yhdistymällä navigoinnista viittaamaansa sisältöön.





*Kuva 6: Esimerkki välilehti-navigaatiosta, valikkorakenteista ja elementtien sijoittelusta mukaillen Steve Krugia*

### 3.3 Lomakkeet

Jukka Korpela (2008, 462) toteaa kirjassaan *CSS verkkosivujen muotoilussa* HTML-lomakeiden olevan itsessään lähellä käyttöliittymää: lomakkeisiin syötetään halutut tiedot, jotka ohjelma käsittelee. Lomakkeiden käytön on oltava helppoa ja tuttua eikä tämän takia Korpelan (2008, 492) mukaan ole syytä muuttaa niiden ulkoasua ratkaisevasti tai poiketa selainten niille antamista oletusasetuksista. Tämä koskee sekä painikkeita että itse lomakkeita.

Lomakkeen tyhjentävä painike voi usein olla turha. Oletusasetuksilla se on samannäköinen kuin lähetä-painike, mikä voi aiheuttaa sekaannusta. Lomakkeen tyhjentävän painikkeen käyttö on perusteltua vain jos lomake on sellainen että saman käyttäjän on tarkoitus lähettää lomake useampaan kertaan. Tällöin painikkeen tyylin on poikettava lähetä-painikkeesta ja sen sopivampi sijoituspaikka on lomakkeen alussa. (Korpela 2008, 470-471.)

Lomakkeiden tulisi olla käytössä äärimmäisen selkeitä myös siksi että niiden täyttäminen koetaan työlääksi. Tämän takia syötekenttien ja vihjetekstien asettelu on perustellusti melko yhdenmukaista eri sivustoilla. Yleisimpiä tapoja asettelussa on käyttää syötekentän oikeaa reunaa lihavoidulle otsikolle ja tasata teksti oikealle, tai asettaa otsikko syötekentän yläpuolelle. Liittyvät vihjetekstit löytyvät kenttien alapuolelta tai oikealta. Tehosteet ja visuaalisuus on hyvä pitää hyvin neutraalina ja antaa käyttäjien keskittyä lomakkeen käyttöön. (Leggett & Maier 2009, 130-131.)

Lomakkeiston yhteenkuuluvat osat kannattaa ryhmitellä omiksi kokonaisuuksikseen rajaamalla ne esimerkiksi div- tai fieldset-elementin sisään. Näistä jälkimmäiselle voidaan määritellä selite legend-elementtiin. Legend-elementillä on omat rajoituksensa CSS-muotoilun suhteen ja se toimii näkyvänä otsikkona fieldset-elementille. (Korpela 2008, 477-479.)

Käyttäjille on hyvä tarjota mahdollisuus täytetyn lomakkeen tulostukselle. Täytetyt lomaketiedot voivat sisältää esimerkiksi tilaustietoja joista käyttäjä haluaa säilyttää kopion. Tässä tehokkain tapa toimia on palauttaa täytetyn lomakkeen tiedot palvelimelta verkkosivuna käyttäjälle jonka tämä voi tallentaa tai tulostaa. (Korpela 2008, 467.)

Käytettävyyttä voi lisätä suurentamalla joitakin lomakkeiden elementtejä. Erityisesti valinta- ja asetusnapit ovat oletusasetuksilla pieniä. Näitä voi Korpelan (2008, 467-468) mukaan suurentaa esimerkiksi aktivoimalla selitystekstin osaksi valintanappia, jolloin valinta tapahtuu myös tekstiä painamalla. Tämä onnistuu HTML-tiedostossa liittämällä label-elementti for-määritteellä input-kenttään. Määritteen for arvon on oltava sama kuin input-kentän id-arvo. (Korpela 2008, 477.)

**Määritellään lomake fieldset-elementtiin ja määritellään lomakkeelle otsake legend-elementillä ja annetaan label-elementille for-arvoksi sama kuin input-kentän id-arvo:**

```
<fieldset><legend></legend></><label for="arvo">LABEL-TEKSTI: <input id="arvo" type="text" name="..." value="..."></label></fieldset>
```

**Ota yhteyttä**

Nimi: \*

Email: \*

Puh: \*

www:

Paikkakunta:

Viesti:

**LÄHETÄ**

\* pakollinen tieto

*Kuva 7: Malli yhteydenottolomakkeesta iDiD.fi –sivustolta. Syötekentät yhdistyvät muodollaan ja aktiivinen alue korostettu hieman kirkkaammalla värillä.*

Vaikka lomakkeistojen tyylimäärittelyihin suositellaan koskemaan mahdollisimman vähän, niin CSS3 mahdollistaa jo pienillä muutoksilla elävöittämään lomakekenttien ilmettä ja istuttamaan ne sopivammin muuhun sivuston ilmeeseen. Sijoittelulla ja mukailemalla käytetyimpiä asetteluja lomakkeisto säilyy toiminnoiltaan tunnistettavana vaikka sen ulkoasua muutetaankin. Harvat ulkoasultaan loppuun asti viimeistellyt verkkosivut käyttävät nykyään lomakkeistoissa ja painikkeissa selainten oletusasetuksia.

## 4 Kosketusnäyttöjen erityispiirteitä

Kosketusnäyttösovelluksia käytetään julkisissa tiloissa kuten rautatieasemilla, messuilla ja myymälöissä. Kosketusnäytöillä on käytettävänä usein yritysten ja tuotteiden esittelyjä tai elämyksellisempiä sovelluksia, kuten pelejä. Ne kaikki markkinoivat palveluaan tarjoavaa yritystä, siksi on tärkeää, että esittelykäytössä olevat sovellukset toimivat tarkoituksessaan ja ovat myös helposti käytettäviä valitulla päätelaitteella.

Dan Saffer (2009, 21-29) käsittelee kirjassaan *Designing Gestural Interfaces* kosketusnäyttöjen ja eleohjauksen erityispiirteitä. Safferin mukaan kosketusnäytöllä hiiren osoitin eli kursori jää usein tarpeettomaksi, koska sormet toimivat osoittimena. Myös erillinen ”hover”-efekti, eli tehoste aktiivisen alueen päällä sormen tai kursorin osoittaessa siihen on useimmiten hyödytön, koska sormea ei juurikaan liu’uteta ruutua pitkin aktiivisia alueita etsien.

Safferin (2009, 26) mukaan pudotusvalikot toimivat harvoin kosketusalustoilla samasta syystä kuin hover-efektikin on hyödytön: sormea ei liu’uteta näytöllä vaan sillä osoitetaan sinne missä toiminnon oletetaan olevan. Pudotusvalikot ovat kuitenkin verkkosivusuunnittelussa yleisiä ja monesti rajallisen tilan vuoksi järkevä tapa esittää alempia valikkorakenteita.

Kosketusalustoilla alasvetovalikot usein pudotetaan alas osoittamalla linkkiä, mikä tuo valikon esiin. Tämän jälkeen valikko pysyy näkyvillä kunnes siitä valitaan haluttu linkki, tai painetaan aluetta valikon ulkopuolella, jolloin alasvetovalikko häviää näkyvistä.

Myös Steve Krug (2006, 110-111) listaa ongelmia alasvetovalikkojen suhteen. Niiden huonoja puolia on se että ne eivät ole kovin hyvin esillä, vaan niitä on etsittävä, eivätkä ole myöskään kovin luettavia.

Ongelmaa aiheuttaa myös se, että alasvetovalikon esiin tuova elementti toimii usein myös linkkinä sisältöön. Useimmilla kosketusnäytöillä selattaessa tätä linkkiä painamalla saadaan alasvetovalikko esiin ja vasta painamalla toisen kerran se toimii linkkinä ja ohjaa eteenpäin. Hiiriohjauksessa pärjätään liu’uttamalla osoitin valikon

päälle, tai suoralla painalluksella sen päällä. Ero ei ole suuri, mutta osa linkeistä voi jäädä huomioimatta jos ne eivät toimi suoraan, tai niiden toiminnassa on ero muihin vastaaviin saman tason linkkeihin nähden.

#### **4.1 Aktiivinen kosketusalue**

Aikuisen sormi on kooltaan noin 16 – 20 mm. Kosketusnäyttöä käytetään yleensä kuitenkin sormenpäällä, mikä on noin puolet edellä mainitusta, eli enimmillään 10 millimetriä. Sormen kärkeä ei kuitenkaan normaalisti käytetä suorassa kulmassa näyttöön vaan vinottain, jolloin se on enemmän kuin pelkkä sormen kärki mutta vähemmän kuin koko sormen leveys, arviolta hieman alle 15 mm. Kynnet ovat käytössä tarkempia mutta ne eivät toimi kaikilla näytöillä. (Saffer 2009, 39.)

Sopivan aktiivisen alueen koon voi laskea pienimmän sormenpään halkaisijan arvon mukaan, jolloin se on 10 mm. Eli tilaa on varattava 10 x 10 mm. (Saffer 2009, 42.) verkkosivuja tehtäessä linkkien leveyttä ei aina pystytä tai edes haluta määrittää, siksi keskityn pelkästään linkin korkeuteen sillä leveys on todennäköisesti tekstilinkeissä käytännössä aina korkeutta enemmän.

Näyttöjen tarkkuudet vaihtelevat paljon joten suositellun 10 x 10 mm:n alueen määrittäminen pikseleinä on hankalaa eri tarkkuuksilla toimivien päätelaitteiden näyttöjen vuoksi. iDiD.fi sivustoa tehdessä olen määrittänyt sen minimiksi 30 pikseliä, laskien tarkkuudella 72 PPI (Pixel Per Inch), eli 72 pikseliä tuumalla, eli noin 28 pikseliä senttimetriä eli 10 millimetriä kohti ja olen pyöristänyt sen 30 pikseliin käyttäessäni kiinteästi määriteltäviä korkeuksia. Tästä arvosta ylöspäin määriteltynä arvo toimii melko hyvin tietokoneiden näytöillä, todennäköisesti kuitenkin aavistuksen suositeltua 10 mm pienempänä.

Tekstin korkeutta määriteltäessä 30 pikseliä on melko korkea arvo, mutta sivusto iDiD.fi sisältää hyvin vähän leipätekstin sisään sijoitettuja linkkejä, jotka saattaisivat olla hyvin lähellä toisiaan sijoittuessaan allekkain ja vaikeasti osoitettavissa sormella, joten riviväli on pienempi.

## 4.2 Käyttäjän oikea- ja vasenkätisyys

Ihmisistä vasenkätisiä on arviolta alle 10 prosenttia, joten on perusteltua optimoida käyttö loppujen 90 prosentin mukaan (Saffer 2009, 39). Web-sovelluksissa on perusteltua pitää käyttöliittymäelementit paikoillaan muuttumattomina (Krug 2006, 60-61), koska käytettävyys voi kärsiä, jos käyttäjälle annetaan mahdollisuus vaihtaa esimerkiksi sisältöalueen ja toisen tason navigoinnin paikkaa. Tämä vaatisi myös käytännössä uudet CSS-määritykset ja oman painikkeen elementtien sijoittelun muuttamiselle ja voi toimia jopa käytettävyyttä heikentävänä tekijänä sekoittaessaan valmista asettelua.

## 4.3 Navigoinnin rakenne ja asettelu kosketusalustoilla

Käsi peittää kosketusnäyttöä käytettäessä osan näytöstä. Dan Saffer (2006, 40-41) pitää perusteltuna sijoittaa perinteisesti ylös asetettavan menurakenteen alareunaan. Tämä toimii tietyntyyppisissä sovelluksissa, mutta web-sivut on syytä pitää yleisesti totuttujen käyttötapojen mukaisena, koska rakenne on todennäköisimmin web-sivuilla pystysuuntainen.

Navigoinnin voi pitää näytöllä kiinteästi tietyssä pisteessä määrittämällä sen sijainnin CSS:llä suhteessa selainikkunaan (`position:fixed;`). Tämä aiheuttaa ongelmia joillakin mobiili- ja sormitietokoneselaimilla (Apple Inc, 2010), mutta on joissakin tapauksissa kierrettävissä eri JavaScript-sovelluksilla.

Valikkojen asemoiminen verkkosivulla suhteessa selainikkunaan on melko harvinaista. Hyvin usein voi olla tarpeen asemoida muita elementtejä suhteessa selainikkunaan `position:fixed` -määrittelyksellä. Navigoinnin paikan muuttaminen muualle kuin suositellusti yläreunaan toimii parhaiten pelkkiin messu- ja julkisiin tiloihin tuleviin esittelysovelluksiin koska verkkosivuja selataan toistaiseksi enimmäkseen normaalinäytöiltä ja hiiri-ohjauksella.

Dan Safferin (2007, 53) kirjan *Designing for Interaction* mukaan käyttöliittymäsuunnittelussakin hyödynnetään usein ”Fittsin lakia”, jonka mukaan suurta ja lähellä olevaa kohdetta on nopeampi ja helpompi osoittaa kuin kaukana olevaa pientä kohdetta. Fittsin lain mukaan tärkeimmät valikkorakenteet on järkevää toteuttaa

siten, että tärkeät kontrollit ovat lähellä käyttäjää ja melko tiiviinä kokonaisuutena. Lisäksi näiden toimintojen tulisi olla tarpeeksi suuria sormella ohjattaviksi. Safferin mukaan myös valikot ja kontrollit kannattaa sijoittaa ruudun reunoille ja kulmiin ja ponnahdusikkunat on hyvä avata lähelle ne esiin tuovaa elementtiä.

Sisällön ja valikkojen jäsentelyä voi Safferin (2007, 53-54) mukaan tarkastella Hickin lain kautta, jonka mukaan käyttäjän on helpompi ja nopeampi valita yhdestä suuresta ryhmästä kuin kahdesta pienestä. Tämän lisäksi käyttäjän toimintaa ja suhtautumista ohjaa myös valikkojen tuttuus käyttäjälle ja toteutettu valikkojen esitystapa, esimerkiksi se käytetäänkö kuvia, sanoja tai multimediaa.

Hickin laista on myös lähtöisin ”seitsemän-sääntö”, jonka mukaan informaatiota tulisi esittää käyttäjälle sen mukaan, mitä ihminen kykenee enimmillään käsittelemään, eli enimmillään noin seitsemän valittavaa kohdetta kerrallaan. (Saffer 2007, 54.)

## **5 Verkkosivusuunnittelu kosketusnäytöille**

### **5.1 Verkkosivun valikkorakenteet kosketusnäytöillä**

Kosketusnäyttöjen erilaiset vaatimukset on hyvä nykyisin huomioida verkkosivusuunnittelussa. Esimerkiksi linkit vaativat lisäkorkeutta, mielellään vähintään aiemmin määritellyt 30 pikseliä. Tässä korostuu kuitenkin verkkosivustoja suunniteltaessa usein vastaan tulevat ongelmat rajallisen käytettävän tilan suhteen. Navigointeja suunniteltaessa on järkevää ryhmitellä sisältöä mahdollisimman tehokkaasti ja pyrkiä pitämään kukin valikko mahdollisimman suppeana ja paljastaa lisää sisältöä vasta linkkien takaa.

Suuret koot saattavat myös häiritä visuaalisia hierarkioita. Visuaalisilla hierarkioilla tarkoitetaan sitä miten tärkeitä asioita korostetaan muusta sisällöstä. Korostuskeinoja ovat esimerkiksi pistekoot, lihavoinnit, väritys, tilankäyttö ja sijoittelu. Näillä korostuksilla voidaan nostaa otsikoita, ryhmittää yhteen kuuluvia asioita ja jäsentää sisältöä. (Krug 2006, 31 – 39.)





*Kuva 8: Web-sovelluksen navigointien kosketusalueiden mukauttaminen kosketusnäytölle huomioiden suositeltu sivurakenne ja kosketusalueet. Vaaleanpunaiset alueet merkitsevät käyttäjän sormenpään kokoa ja ne ovat toiminnallisten alueiden päällä. Toiminnallisina esimerkkeinä on kirjautumis-toiminto, haku-toiminto, navigoinnin eri tasot, lisävalikko ja linkki leipätekstissä.*

Linkit tekstin seassa eivät ole ongelma, ellei niitä ole tai osu useampaa allekkain. Normaalisti rivikorkeus määriteltynä CSS:ään line-height-komennolla, on alle 20 pikseliä leipätekstillä. (Martin 20.4.2009.)

Valikkorakenteen ovat CSS- ja HTML-määritysten kautta muokattavissa mukautumaan kosketusnäyttöjen suositeltuihin vaatimuksiin. Isot linkkien koot ovat haasteellisia

tilankäytön suhteen, mutta suositeltua kokoa pienempiäkin toiminnallisia elementtejä voi käyttää näyttölaitteen rajoitusten mukaan. (Saffer 2009, 43.)

Kuvakkeiden eli ikonien käyttö voi olla tehokasta, mutta niitä ei saa olla liikaa. Käytettävät ikonit on valittava tarkkaan ja on hyvä tietää ainakin jotakin kohdeyleisöstä ikonien järkevän valinnan kannalta (Fadeyev 2009). Kosketusnäyttöillä ikonit voivat olla erityisen tehokkaita linkkien apuna, sillä ikonit vaativat melko runsaasti tilaa jotta ne voi esittää toimivasti, käytön vain on oltava perusteltua. Steve Krug (2006, 102-103) suositteleeikin käyttämään niin paljon tilaa, kuin on tarpeen, mutta ei sen enempää. Eli tyhjää tilaa ei kannata viljellä liikaa elementtien ympärille. Tyhjä tila voi toimia myös tyyllillisenä tehokeinona.

Leggett ja Maier korostavat kirjassa *The Smashing Book – Usability Principles for Modern Websites* että alavetovalikkojen käytön on oltava harkittua ja parhaimmillaan ne ovat enimmillään kahdessa tasossa toimivissa navigoinneissa. Niitä käytettäessä on käyttäjälle selvästi kerrottava valikossa minkä linkkien takaa löytyy lisävalintoja.

Alasvetovalikkojen etu on kuitenkin niiden vaatima pieni tilan. Alasvetovalikkoja tulisi kuitenkin käyttää vain kuin ehdottomasti tarve vaatii, sillä käyttäjät näkevät niistä ainoastaan yhden valinnan kun alavetovalikko ei ole aktiivinen. Pudotusvalikkoihin ei saa liittää liikaa valintoja ja niiden käytettävyys sivustolla on syytä testata ennen niiden valitsemista navigoinniksi. (Leggett & Maier 2009, 128-130.)

Toimivissa ratkaisuissa alavetovalikot toimivat hieman ponnahdusikkunan tavoin. Valikkojen ollessa isoja niihin pystyy lokeroimaan asiakokonaisuuksia ja ne sallivat tehokkaammin ikonien ja kuvien käytön vihjeistyksessä. (Leggett & Maier. 2009, 128-129.)

## **5.2 iDiD.fi verkkosivusto ja kosketusalustakäyttö**

iDiD.fi on Connecting Talents Oy:n älynäyttötuotteita ja -palveluja esittelevä verkkosivu. Se on rakennettu iStart-julkaisujärjestelmälle ja sen toteutuksessa on pyritty huomioimaan kosketusnäyttöjen käytön erityisvaatimuksia: Linkeille on varattu reilusti kosketusalaa, teksti on isoa ja selkeää. Leipätekstin riviväli ja -korkeus on pidetty väljänä ja otsikointi on erotettu taustavärillä muusta tekstistä.

Valikkoratkaisussa alasetoalikko oli perusteltua koska visuaalisesti päänavigoinnista oli tarkoitus tehdä vahva ja selvästi erottuva elementti kolmella pääsivulla.

Alasetoalikko sallii myös alasetujen lisäämisen siten, että päänäkyä pysyy neutraalina ja valinnoiltaan suppeana.

Sivuston tunnus toimii myös linkkinä aloitussivulle, samoin kuin navigoinnin ensimmäinen linkki. Sivuston leipätekstin sisään sijoittuvat linkit on erotettu muusta tekstistä ja navigoinnista omalla värillä.

Väriarinnat, otsikointi ja tyyli sisällön asettelussa mukailevat muuta iDiD-markkinointimateriaalia.



Kuva 9: iDiD.fi -verkkosivun yhteydenotto-sivu. Toiminnalliset elementit korostettuina punaisella läpinäkyvällä ympyrällä joka kuvaa määrittelyäni 30 pikselin kosketusala linkille kosketusnäytöillä.

## 6 Mukautuvan verkkosuunnittelun keinot

Mukautuvassa verkkosuunnittelussa huomioidaan verkkoselaukseen käytettävät erilaiset näyttölaitteet. Monet mobiiliselaimet piirtävät verkkosivut ja huomioivat HTML-sivujen CSS-määrittelyt kokonaan tai lähes samoin kuin ”täysiveriset” verkkoselaimet joita käytetään perinteisemmällä tietokoneilla. Mobiiliselaimet piirtävät sivut huomattavasti pienempänä näytölleen, ellei niille ole määritelty erillistä CSS-tyyliä.

Koska verkkosivuilla esitetään eniten kuvaa ja tekstiä ja nämä asettuvat suhteessa määrittelyihin ympäröiviin rakenteisiin, niin käsittelen näiden elementtien, eli kuvien, tekstin ja niitä ympäröivien rakenteiden eri laitteille mukautuvaa CSS-tyylimäärittelyä.

Verkkosivuja selataan jopa pelikonsoleilla, joissa osassa löytyy verkkoselain. Uusia laitteita tulee käyttöön nopeasti ja suunnittelussa huomioitavia selaimia alkaa olla valtava määrä. Tässä pisteessä on järkevää miettiä, mille kaikille alustoille verkkosivujen tyylejä on järkevää määritellä erikseen. (Marcotte 2010). Ethan Marcotte esittelee A LIST apart -verkkosivuilla artikkelissaan *Responsive Web Design* vaihtoehdoisen lähestymistavan verkkosuunnitteluun jossa huomioidaan useat eri päätelaitteet. Ideana on skaalata sivuja ja sisältöä kun päätelaitteen näytön tarkkuus täyttää tietyn määritellyn pikselileveyden tai resoluution.

Verkkosivuja selatessa käyttäjät harvoin pitävät selainikkunaa suurimmassa mahdollisessa koossa, vaikka sivuja luettaisiinkin isolta tietokoneen näytöltä. Päätelaitteet myös käyttävät joko vaaka- tai pystysuuntaista asettelua ja joskus sallivat vaihtamisen näiden välillä. Mukautuva verkkosivusuunnittelu huomioi tämän ja sivut asettuvat eri päätelaitteille suunnittelijan tarkoittamalla tavalla. Kuvat, teksti ja muu sisältö asettuvat näytölle luettavina ja vähentävät käyttäjän tarvetta tarkentaa tekstiin ja selata kuvia eri suuntiin. Parhaimmillaan kaikki sisältö asettuu pienienkin päätelaitteiden näytölle helposti tarkasteltavaan ja nopeasti luettavaan muotoon. (Knight 2011.)

## **6.1 Päätelaitetunnistus ja käytettävän CSS-tyylin määrittäminen**

Päätelaitetunnistuksella määritellään verkkopalvelua käyttävän päätelaitteen tyyppi. Tätä käytetään jotta laitteen ominaisuudet voidaan huomioida ja tarjota sille halluttaessa sopivinta tyylimäärittelyä. Päätelaitetunnistusta käytetään, koska erityisesti mobiililaitteet käyttävät hyvin erilaisia resoluutioita ja pystyvät hyödyntämään vaihtelevasti HTML-sivuilla käytettäviä komponentteja. (Päätelaitetunnistus 2009.)

Perinteisemmät selaimet voivat vaatia joissakin tapauksissa omaa tyylimäärittelyä. Tällöin käytetään vastaavasti toimivaa selaintunnistusta. Päätelaitetunnistuksella voidaan vaikuttaa käyttäjälle näkyvään sisältöön ja tapaan jolla se esitetään, sillä siis pyritään optimoimaan käyttökokemusta ja käytettävyyttä. (Päätelaitetunnistus 2009.)

Päätelaitteen voi ohjata haluttuun osoitteeseen JavaScriptillä, PHP:llä tai lisäämällä uudelleenohjaus .htaccess-tiedostoon. Näin voi myös tarjota käyttäjälle mahdollisuuden tarkastella myös muuta versiota sivustosta. (Mike, 2010.)

Mukautuvalla HTML-sivujen CSS-tyylimäärittelyksellä on tarkoitus vähentää juuri edellä mainittujen monien erillisten päätelaitekohtaisten tyylitiedostojen tai alisivustojen tarvetta.

## **6.2 Päätelaitteen näyttöön mukautuva CSS-määrittely**

CSS3 sallii tyylitiedostojen linkityksessä automaattisesti "kysyä" käytettyä päätelaitteen näytön tarkkuutta ja tuoda tyylejä ehtojen täytyessä (Knight, 2011). Kyselyyn käytetään Media Query -tiedoksiantoja, joka tehdään head-tagien sisään HTML-tiedostoon. Tätä käytetään määritellessä miten verkkosivujen sisältö näkyy esimerkiksi näytölle tai tulostettaessa sivun sisältö paperille.

Esimerkiksi päätelaitteen näytön leveyden ollessa enimmillään 480 pikseliä (480 pikseliä on sama kuin iPhone3:n ja iPhone3GS:n leveys vaakasuunnassa) tuodaan käyttöön tyyli stylesmall.css. Näytön tarkkuuden ollessa suurempi tyyli stylesmall.css ohitetaan.

## **Tehdään viittaus haluttuun tyylitiedostoon näytön koon mukaan:**

```
<link rel="stylesheet" media="screen and (max-device-width: 480px)
"href="stylesmall.css" />
```

Viittaus voidaan sisällyttää myös määriteltyn päätyylitiedostoon ja tuoda esiin @import-komennolla, jolloin HTML-tiedosto säilyy siistimpänä. Viittaus head-tagtiin määritellyn CSS-dokumentin sisältä edelleen linkitettyyn CSS-tiedostoon tapahtuu seuraavasti:

```
@import url("stylesmall.css") screen and (max-device-width:480px);
```

Tyylimäärittely päätelaitekohtaisesti onnistuu myös CSS-tiedoston sisällä seuraavasti:

```
@media screen and (max-device-width:480px); { selektori { attribuutti:attribuuttiarvo;
} }
```

Tällä voi tehostaa ja selkeyttää jonkin verran tyylimäärittelyjä, koska tyylimäärittelyjä saattaa yhtä sivustoa kohti kertyä niin paljon että melko yleiseksi tavaksi on tullut jaotella esimerkiksi yleinen rakenne, typografia ja navigointi omiksi tyylitiedostoikseen suunnittelijan ratkaisuksista riippuen.

Edellä mainitut ohjeet mukailevat A LIST apart -sivustolla julkaisun Ethan Marccotten kirjoittaman artikkelin *Responsive Web Design* neuvoja.

## **6.3 CSS-tyylien nollaus**

Html-rakenteita määriteltäessä on joskus syytä nollata kaikki tyylitiedot. Nollaus on perusteltua, koska eri selaimet käyttävät eri oletusasetuksia. Määrittelemällä asetukset kaikille samaksi, varmistetaan tyylimäärittelysten näkyminen yhdenmukaisesti kaikilla selaimilla. Tällä myös pystytään varmistamaan se, että halutessa voidaan itse määrätä mitä tyyliä tai kokoa elementtien tehosteissa käytetään. (Meyer & Meyer, 2007.)

Käsittelen HTML- ja CSS-esimerkkejä pääosin CSS3:n ja HTML5:n mukaisesti, mutta siten, että määrittelyt näkyvät samoin myös HTML5 tukemattomilla selaimilla.

HTML5 tuo mukanaan joitakin uusia attribuutteja ja elementtejä.

### **Nollataan haluttujen attribuuttien CSS-tyylitiedot:**

```
# selektori, . selektori, selektori { margin:0; padding:0; border:0; font-size:100%;  
font:inherit; vertical-align:baseline; }
```

HTML5:tä tukemattomat selaimet antavat tunnistamattomille elementeille arvon *display:inline;*. Antamalla arvon *display:block;* HTML5-elementeille, myös HTML5:tä tukemattomat selaimet näyttävät sivun kuten HTML5:tä tukevat selaimet. (Pilgrim 2010, 3.)

### **Määritellään HTML5-elementeille oikea arvo niille selaimille jotka eivät tue HTML5:tä:**

```
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {  
display:block; }
```

### **Poistetaan näkyvät listojen ja lainausten tyylit:**

```
ol, ul { list-style:none; }
```

```
blockquote, q { quotes:none; }
```

```
blockquote:before, blockquote:after, q:before, q:after { content:""; content:none; }
```

### **Nollataan taulukkorakenteet ja poistetaan niiltä näkyvä rajausta:**

```
table { border-collapse:collapse; border-spacing:0; }
```

### **Poistetaan aktiivisten linkkien ympäriltä yleensä pistemäisenä reunuksena näkyvä rajausta:**

```
a { outline:0; }
```

Verkosta on myös ladattavissa valmiita CSS-nollaustiedostoja.



## 6.4 Mukautuvat rakenteet

Verkkosivuja suunniteltaessa rakenteiden määrittelemisen kiinteään levyisiksi on usein helpompaa, koska näin voidaan määritellä helposti miltä sisältö näyttää oli päätelaitteen tarkkuus mikä tahansa. Kiinteään kokoon suunniteltaessa elementtien koot ja etäisyydet pysyvät pikseleinä aina samoina mutta tällainen suunnittelu ei toisaalta hyödynnä tilaa, mitä suuremmilla näytöillä selainikkunat saattavat tarjota. (Fadeyev 2009, 30-33.) Kiinteään kokoon suunniteltaessa myöskään pienten päätelaitteiden rajoituksia ei välttämättä voida huomioida kovin hyvin.

Määrittelemällä rakenteet joustaviksi (fluid-, liquid-layout) voidaan tehokkaimmin hyödyntää päätelaitteen näyttötilaa. Suunniteltaessa rakenteita voidaan esimerkiksi emoelementille määritellä prosenttiosuus selainikkussa näkyvästä tilasta ja tämän sisälle asetettaville elementeille taas osuus prosentteina emoelementistä. Rakenteiden suunnittelu voi vaatia tarkempia laskusuorituksia, mutta käytettävissä oleva tila voidaan hyödyntää paljon tehokkaammin kuin tarkoilla pikseliarvoilla määriteltäessä. (Fadeyev 2009, 36-38.)

Emoelementeille ja tekstikentille on kuitenkin hyvä määrittää suurimpia sallittuja arvoja, koska pitkät tekstirivit eivät ole kovin luettavia (Fadeyev 2009, 38-39) ja tällä voi estää ulkoasun hajoamista liikaa määritelystä. Sivuston suurimman uloimman elementin leveyden määrittämiseen voi käyttää em-arvoa, tällöin selaimelle em-määritelty teksti elää samassa suhteessa kaikkien em-määriteltyjen elementtien kanssa. Elementtien marginaalit ja sisennykset voi määritellä prosenttiarvoina, näin selainikkunan tai päätelaitteen koon muuttuessa määritelty etäisyydet ja haluttu tyhjä tila säilyvät. (Fadeyev 2009, 42.)

**Määritellään uloimmille elementeille pienin leveys sekä leveys asettumaan näytön tai selainikkunan leveyden mukaan.**

```
selektori { margin:0 auto; min-width:480px; max-width:100%; }
```

Tehokas CSS-määrittely saattaa siis vaatia eri tekniikoiden käyttöä kiinteistä pikselimäärittelyistä elastisempiin em- ja prosenttimäärittelyksiin. Vaikka rakenteet ja sisältö mukautuisivat päätelaitteen näytön tai selaimen tilan mukaan niin lähes aina

bannerit määritellään tiettyihin standardikokoihin pikseleinä. Järkevintä on siis määritellä suurin ja pienin haluttu koko, jotka säilyvät oli päätelaitteen tai selaimen tarkkuus mikä tahansa. Samat määrittelyt korkeuksille saattavat olla myös tarpeellisia. (Fadeyev 2009, 57-61.)

Elävät rakenteet ovat suunnittelijalle haastavampia käyttää ja ratkaisu valittavan tekniikan, eli kiinteän ja joustavamman rakenteen välillä kannattaa tehdä sen mukaan mitä sisältöä sivustolla tullaan esittämään. Ainakin osan sisällöstä voi olla järkevää esittää siten, että se huomioi verkkosivuun käytettävän päätelaitteen rajoitukset.

Joustavien rakenteiden yhdistäminen CSS3:n mahdollistamiin visuaalisiin tehosteisiin voi vähentää kuvien käytön tarvetta taustoissa ja tehosteissa. Tämä myös keventää verkkosivua sekä sallii siistimmät kokomuutokset. Huomioitavaa tosin on eri selainten rajoittunut tuki CSS3:een ja siten oma ja verkkosivun tilaajan valmius kompromisseihin eroissa ulkoasussa selaimen muuttuessa.

## **6.5 Tekstin CSS-määrittely**

Teksti on CSS:ssä järkevää määritellä pikseleiden (px) sijaan em-arvolla. Näin teksti muuttaa kokoaan paremmin eri selaimilla. (Rutter, 2007.) Em-pohjaisen tekstien CSS-suunnittelun vahvuus on siinä, että käyttäjän suurentaessa tekstikokoa selaimessaan otsikkojen ja leipätekstin suhteet sekä määritellyt kokoerot säilyvät koska em-määritellyt elementit saavat arvonsa emoelementin koon kautta. (Cattaneo, Peters & Tan 2009, 74-75.) Arvo 1 em vastaa arvoa 100% (Meyer 2006, 111).

Selaimen oletusasetuksena fontin eli kirjasimen koko on 16 px ja sen muuttaminen muotoon em tapahtuu määrittämällä fonttikoko joko body-elementille muotoon em ja hakemalla kertoimet tekstille siitä, tai antamalla elementille body tekstin kooksi 100% ja muuttamalla tekstielementtejä suhteessa tähän lukuun. 10 pikseliä leipätekstille saadaan siis määrittelemällä body-elementille 100% ja p-elementille 0.625 em, tai antamalla body-elementille suoraan arvo 0.625 em. (Cattaneo, Peters & Tan 2009, 81-87.)

Ero on siinä, että joskus on helpompaa antaa tekstielementeille arvoja suhteessa pyöreään lukuun 10 px, joka oli määritelty body-elementille ja laskea esimerkiksi kertoimella 1.4 em halutulle tekstille kooksi 14 pikseliä.

**Määritellään elementin body kautta tekstille koko suhteessa oletusasetukseen 16px, rivikorkeus suhteessa body-elementille määritettyyn fontin-kokoon ja alamarginaali elementille p.**

```
body { font-size:0.875em; line-height:1.5em; } /* 0.875 x 16 = 14 px */
```

```
p { margin-bottom:1.5em; }
```

Nyt fontin koko on 14 pikseliä, rivikorkeus 21 pikseliä ja leipätekstin tagin alamarginaali 21 pikseliä.

Joillakin vanhemmilla Safari-selaimilla saattaa olla tulla vastaan kokoeroja tekstin kanssa tietyillä tekstin tyylimäärittelyillä em-koon määrittystä käytettäessä, mutta ongelma on melko marginaalinen. Laajempi ongelma on vielä melko laajalti käytettyjen Internet Explorer 6 ja erityisesti – 7 versioiden kanssa, jotka näyttävät rivikorkeuden tietyillä tehoste-, tai kokotyyleillä väärin. Tätä voi kiertää määrittelemällä esimerkiksi poikkeustyylin Internet Explorer 8:aa vanhemmille IE-selaimille (Rutter 2007.)

Internet Explorer 8:aa (IE 8) vanhempien selainten tyylien korjaus onnistuu esimerkiksi viittaamalla ehtolausekkeella IE 8:aa vanhempiin selaimiin ja määrittelemällä näille haluttu korjaus omalla tyylillä CSS-tiedostoon.

```
<head><!--[if lte IE 8]>--> viittaus tyylitiedostoon <!--<[endif]></head>
```

## 6.6 Kuvien koon CSS-määrittely

Ethan Marcotte esittelee verkkoartikkelissaan *Fluid Images* (Marcotte, 2010a) tavan antaa sisällössä käytetyille kuville suurimmaksi leveys arvoksi 100% (`img{ max-width:100%; }`) ja vanhempien Internet Explorer –selainten puutteellinen tuki `max-width`-määrittystä kohtaan korjataan määrittelemällä kuvalle leveys 100% (`img{ width:100%; }`) omaan tyylimäärittelykseen. Tällä määrittelyllä kuva ei ylitä alkuperäistä pikselileveyttään vaikka elementti minkä sisään se on asetettu olisi leveydeltään

enemmän, kuin kuvan leveys on pikseleinä. Näin kuvaa ympäröivän elementin leveyden ollessa pienempi kuin kuvan alkuperäinen leveys on pikseleinä näkyy kuva yhtä leveänä kuin ympäröivä elementti.

Kuvaa pienennettäessä sen laatu kuitenkin saattaa kärsiä. Tätä voi korjata JavaScriptillä vaikka käyttöjärjestelmäkohtaisesti muuttamalla tapaa jolla kuvan pikselimäärän vähetessä kuva esitetään. Määrittämällä kuville kuvakoon pienentyessä interpolointiarvoksi bicubic laatu pysyy parempana (`img { -ms-interpolation-mode: bicubic; }`). (Marcotte 2010a.)

### **Määritellään kuville suurin leveysarvo ja Windows-käyttöjärjestelmälle interpolointi tyyppi.**

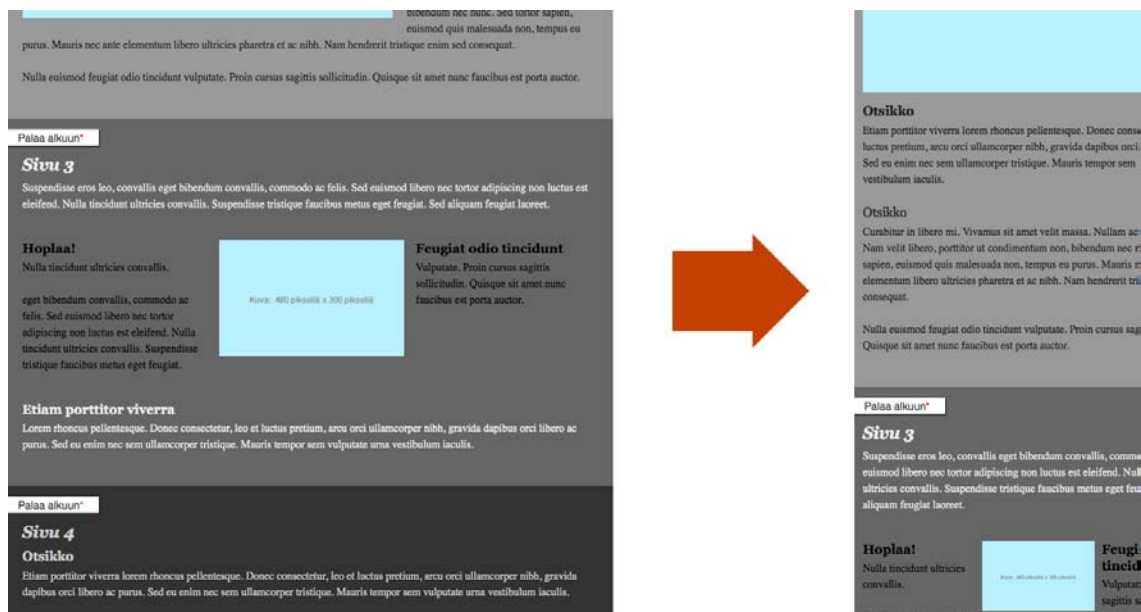
```
img { max-width: 100%; -ms-interpolation-mode: bicubic; }
```

Kuvien leveydet asettuvat nyt suhteessa ympäröivään elementtiin, mutta ne näkyvät enimmillään yhtä isona kuin kuvien alkuperäinen leveys on pikseleinä. Korkeus muuttuu automaattisesti, mutta sen voi varmistaa `height:auto` -määrittelyksellä.

Kuvat mukautuvat ympäröivään rakenteeseen kun taas kuvia ja tekstiä ympäröivät rakenteet määrittyvät haluttaessa päätelaitteen näytön koon mukaan.

## 7 Mukautuva verkkosuunnittelu käytännössä

Mukautuvan verkkosuunnittelun hyödyt näkyvät ehkä parhaiten paljon käytetyillä sivustoilla joita selataan paljon erilaisilla päätelaitteilla ja joiden sisältö on tekstipainotteista. Käytettävyys lisääntyy kun sisältö asettuu suoraan suunnittelijan parhaaksi katsomalla tavalla jättäen käyttäjälle kuitenkin mahdollisuuden muuttaa tekstin kokoa. Myös kuvat asettuvat oikein suunniteltuna suhteessa ympäröivään elementtiin eivätkä vie tilaa tekstiltä näin haluttaessa.



*Kuva 10: Mukautuvassa verkkosuunnittelussa selainikkunaa pienennettäessä emoelementin leveyden prosenttiosuuksina määritellyt sisältöalueet kapenevat ja teksti asettuu uudelleen luettavampaan muotoon. Myös kuvat pienenevät suhteessa emoelementtiin, mutta eivät ole milloinkaan suurempia kuin niiden alkuperäinen pikselileveys on.*

Suunnittelutyössä tarkkuutta vaaditaan enemmän mutta verkkosivustoista voi tehdä kevyempiä ja pitkäikäisempiä. Toisaalta huolellinen suunnittelu voi vähentää tarvetta monille päätelaitekohtaisille CSS-dokumenteille.

# Sivuston tunnus\*

**Sivu 1**   **Sivu 2**   **Sivu 3**   **Sivu 4**

## Sivu 1

Aliquam varius lacinia lectus, viverra vehicula lorem ultricies ut. Aenean ornare ligula auctor tellus pretium ullamcorper. [Etiam convallis lacus dictum magna commodo](#) quis pulvinar lectus dapibus. Donec semper egestas nisi quis feugiat. Suspendisse mollis, enim porttitor scelerisque accumsan, nunc lacus mollis est, sit amet ullamcorper est felis in lacus. Suspendisse eros leo, convallis eget bibendum convallis, commodo ac felis. Sed euismod libero nec tortor adipiscing non luctus est eleifend. Nulla tincidunt ultricies convallis. Suspendisse tristique faucibus metus eget feugiat. Sed aliquam feugiat laoreet.

Palaa alkuun\*

## Sivu 2

Kuva: 480 pikseliä x 300 pikseliä

### Otsikko

Etiam porttitor viverra lorem rhoncus pellentesque. Donec consectetur, leo et luctus pretium, arcu orci ullamcorper nibh, gravida dapibus orci libero ac purus. Sed eu enim nec sem ullamcorper tristique. Mauris tempor sem vulputate urna vestibulum iaculis.

### Otsikko

Curabitur in libero mi. Vivamus sit amet velit massa. Nullam ac iaculis risus. Nam velit libero, porttitor ut condimentum non, bibendum nec nunc. Sed tortor sapien, euismod quis malesuada non, tempus eu purus. Mauris nec ante elementum libero

ultricies pharetra et ac nibh. Nam hendrerit tristique enim sed consequat.

Nulla euismod feugiat odio tincidunt vulputate. Proin cursus sagittis sollicitudin. Quisque sit amet nunc faucibus est porta auctor.

Palaa alkuun\*

## Sivu 3

Suspendisse eros leo, convallis eget bibendum convallis, commodo ac felis. Sed euismod libero nec tortor adipiscing non luctus est eleifend. Nulla tincidunt ultricies convallis. Suspendisse tristique faucibus metus eget feugiat. Sed aliquam feugiat laoreet.

### Hoplaa!

Nulla tincidunt ultricies convallis.

eget bibendum convallis, commodo ac felis. Sed euismod libero nec tortor adipiscing non luctus est eleifend. Nulla tincidunt ultricies convallis. Suspendisse tristique faucibus metus eget feugiat.

Kuva: 480 pikseliä x 300 pikseliä

### Feugiat odio tincidunt

Vulputate. Proin cursus sagittis sollicitudin. Quisque sit amet nunc faucibus est porta auctor.

### Etiam porttitor viverra

Lorem rhoncus pellentesque. Donec consectetur, leo et luctus pretium, arcu orci ullamcorper nibh, gravida dapibus orci libero ac purus. Sed eu enim nec sem ullamcorper tristique. Mauris tempor sem vulputate urna vestibulum iaculis.

Palaa alkuun\*

## Sivu 4

### Otsikko

Etiam porttitor viverra lorem rhoncus pellentesque. Donec consectetur, leo et luctus pretium, arcu orci ullamcorper nibh, gravida dapibus orci libero ac purus. Sed eu enim nec sem ullamcorper tristique. Mauris tempor sem vulputate urna vestibulum iaculis.

[Palaa alkuun](#)

[Sivu 1](#)

[Sivu 2](#)

[Sivu 3](#)

[Sivu 4](#)

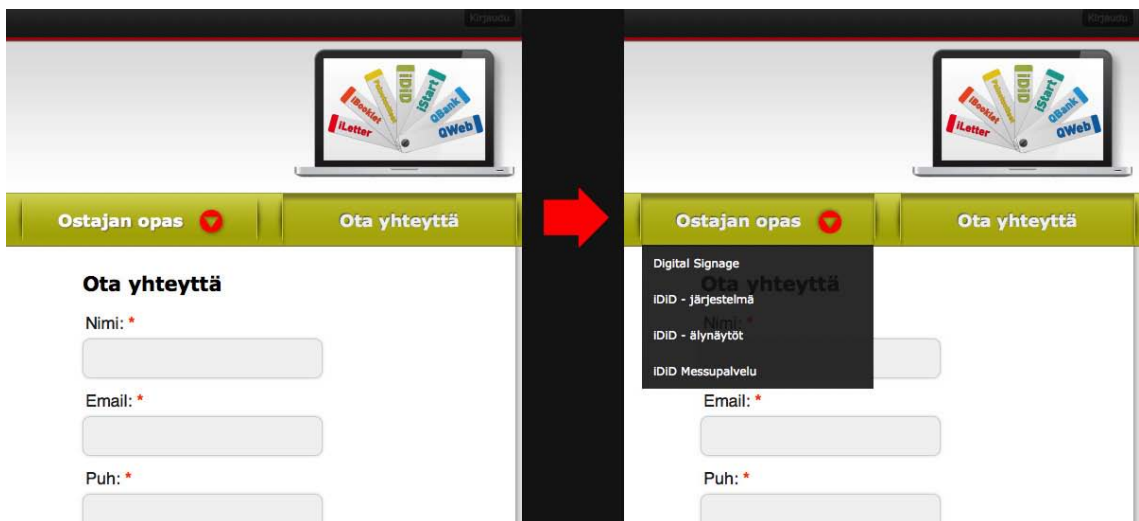
Opinnäytetyö :: Mukautuva verkkosuunnittelu :: Antti Tolppanen 2011

*Kuva 11: Pystysuuntaan rakennettu mallisivu jossa 480 pikseliä leveää ja 300 pikseliä korkeaa kuvaa on käytetty ympäröiviin elementteihin mukautuvasti. Navigoinnin toiminnallisten elementtien korkeus on yli 30 pikseliä ja navigointi tapahtuu ankkureihin joiden läheisyyteen on lisätty alkuun palauttava toiminto.*

## 8 iDiD-verkkosivuston kehitysehdotuksia

### 8.1 Navigointi

Alasvetovalikot vaativat useamman painallukseen niissä edettäessä, mikä vähentää käytettävyyttä. Päälinkki jonka alla on alasvetovalikko, vaatii siis toimiakseen linkkinä kaksi kosketusta ja tämä voi hämätä käyttäjää. Sivustolla iDiD.fi valikot eivät kuitenkaan ole koskaan kahta tasoa syvempiä, jolloin alasvetovalikkoa voi pitää vielä toimivana ratkaisuna koska kosketus alustojen muut vaatimukset on huomioitu ja sivusto on sisällöltään suppea, muutaman alasivun kokonaisuus.



*Kuva 12: Alasvetovalikon esiin tuova valikko toimii myös linkkinä, mutta normaalisti kosketusalustoilla käytettynä ensimmäinen kosketus tuo alemman valikon esiin ja toinen kosketus toimii linkkinä. Kuvassa korjattu punaisella värillä vihje käyttäjälle sen takaa löytyvästä valikosta.*

Vaihtoehtoisia tapoja esittää alatasen linkkejä on vihjeistää käyttäjää pudotusvalikosta esimerkiksi pienellä nuolella painikkeen yhteydessä. Tämä kertoo että sen takana on alasvetovalikko, tai että toiminto eroaa muista painikkeista. Valikoissa joissa alasivuja ei ole kovin montaa, voi liikuttaa sivustoa sivusuunnassa esimerkiksi jQuery-liitännäisten avulla. Ongelma on se että aivan kaikki julkaisujärjestelmät eivät

välttämättä salli tällaisten liitännäisten käyttöä. Pienemmissä projekteissa niiden käyttöä saattaa myös vähentää joidenkin liitännäisten maksullisuus kaupallisissa sovelluksissa.

## 8.2 Linkit

Footer-osan yläreunassa olevat muita tuotteita esittelevät linkit voidaan kokea hämääviksi koska ne yhdistyvät väriyksellään päänavigointiin, eikä niissä ole vihjeistystä siitä, että ne vievät ulos sivustolta.

Vaihtoehtoisia tapoja esittää nämä linkit olisi ohjata ne footer-osioon navigoinnin pienen version oikealle puolelle otsikoituna, tai kertoa niiden läheisyydessä tekstillä niiden ohjaavan ulos sivustolta.



*Kuva 13: Sivustolta ulos vievät linkit esitettynä punaisella footer-osassa sivustolla idid.fi.*



*Kuva 14: Sivustolta ulos vievät linkit otsikoituna vihjetekstillä.*



### 8.3 Pystysuuntaan jäsennetty verkkosivu

Verkkosivuston voi rakentaa jäsentämällä sivuston sisällön allekkain yhdelle sivulle ja asettaa haluttuihin kohtiin sivua ankkureita, joihin navigointi ohjaa. Tämä toimii erityisesti suppeampaa sisältöä esittävillä kampanjasivuilla ja yritysten esittelysivuilla. Kovin laajaa sivustoa tällaisella ratkaisulla ei ole helppoa pitää selkeänä ja hallittavana.

Pystysuunnassa rakennetussa ratkaisussa hyviä liitettäviä lisätoimintoja ovat takaisin sivun alkuun vievät linkit footerissa ja footer-osan apunavigointi linkkeineen ja käytetyimpine toimintoineen. Myös navigoinnin tulisi seurata jollakin tapaa sivua alaspäin selattaessa, tai liittää ankkurikohtiin navigoinnin toimintoja kuten takaisin navigointiin tai sivuston alkuun ohjaavan toiminnon.

Pystysuuntaan rakennetuissa sivuissa siirtymissä sivun osioista toiseen animoinnit helpottavat hahmottamaan siirtymän konkreettisuutta. Se auttaa myös hahmottamaan mihin suuntaan sivustolla edetään ja missä ollaan linkin painamisen jälkeen. Osioihin voi myös jättää käyttäjälle vihjeitä mistä suunnasta lisäsisältöä löytyy.

Kosketusnäytöille pystyrakenne sopii siirtymät animoituina linkeistä ankkureihin erityisen hyvin, koska siirtymä mukailee tapaa jolla sivustolla liikutaan sormella sitä selattaessa ja käyttäjä näkee minne linkin kosketuksen jälkeen siirrytään ja tämä auttaa hahmottamaan missä kohtaa sivustoa ollaan.

Lisätoimintoja voi kätkeä hyvin vihjeistettynä myös erillisten esiin tulevien liukuvalikoiden taakse. Nämä näkyvät samoin kuin alasvetovalikot, mutta eivät toimi itse linkkeinä ja sisältävät usein vain yhden toiminnon kuten yhteydenottolomakkeen. Liukuvalikot tulevat esiin joko osoittimen siirtyessä päälle tai kosketuksesta ja niitä voi asettaa välilehtien tavoin esimerkiksi footeriin. Myös näissä on hyvä käyttää animointia esiintuomiseen.



Kuva 15: Sivuston rakenne esitetty pystysuunnassa. Navigointi ohjaa ankkureihin ja kaikki lisätoiminnot liitetty "sivuihin" tai footeriin. Alkuun palauttavan toiminto kulkee joko navigoinnin mukana, tai sen voi liittää ankkurikohtiin. Jälkimmäinen on rajallisen tilan vuoksi toimivampaa pienillä päätelaitteilla sivua selattaessa.

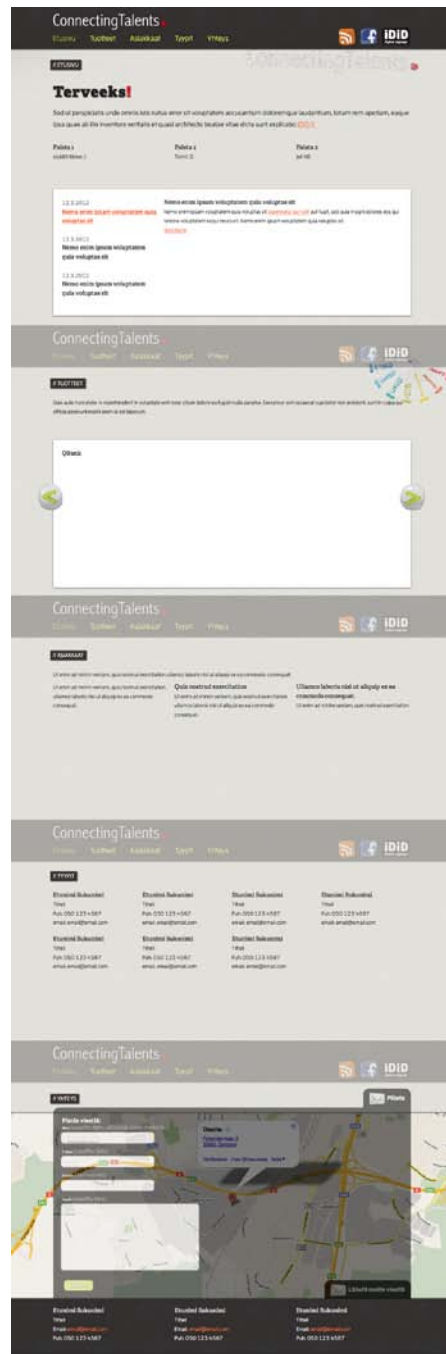
## 9 Yhteenvetoa

Vaikka verkosta löytyy valmiita nollauspohjia CSS-tyyleille ja valmiiksi määriteltyjä runkoja CSS-rakenteille, niin niiden tehokas hyödyntäminen vaatii hyvät perustiedot CSS:n ja HTML:n käytöstä, sillä valmiisiin CSS-pohjiin tehdyt muutokset saattavat vaikuttaa yllättävästi koko rakenteeseen. Vaikka parhaimmillaan valmiit CSS- ja HTML-rungot ovatkin hyvin dokumentoituja, niin koen varsinkin pienemmissä sivustoissa monesti mielekkäämpänä koostaa rakenteet alusta loppuun itse tai käyttäen omia runkopohjia selvyiden vuoksi. Tämän takia olen koostanut opinnäytetyössäni eri lähteiden neuvoja yhteen.

Tämä opinnäytetyö on ollut myös perehtymistä CSS- ja HTML-määrittelyihin ja verkkosivun rakenteissa tehtyjen ratkaisujen perustelemista, mikä on tehostanut omaa työskentelyäni ja selkeyttänyt erityisesti em-pohjaisten määrittelyjen toimivuutta ja tehokkuutta.

Opinnäytetyössäni esittelemiäni tekniikoita olen hyödyntänyt koostaessani sivupohjia iDiD.fi:tä seuraaviin projekteihin. Vaikka verkkosivun tilaajien antamat toiveet ohjaavat tai rajoittavat käytettyjä tekniikoita ja ratkaisuja, niin suurimpana vaikuttajana pidän kuitenkin valitun tai tilaajalle tarjottavan julkaisujärjestelmän rajoituksia tai mahdollisuuksia.

Monille eri päätealustoille suunniteltaessa tehokkaimpana ratkaisuna olen päätenyt käyttämään venyviä ja emoelementtien kokoon mukautuvia rakenteita, em-pohjaista typografiamäärittelyä, isoa tekstin ja linkkien kokoa ja riviväliä, sekä jQueryn ja JavaScriptin käyttöä tarvittaessa. Näiden lisäksi pyrin pitämään verkkosivun rakenteen mahdollisimman perinteisenä. Nämä edellä mainitut seikat voi yhdistää tehokkaasti esimerkiksi pystyrakenteiseen sivustoon, jonka voi halutessaan toteuttaa vain yhtä HTML-tiedostoa käyttäen.



Kuva 16: Pystysuuntaan rakentuva verkkosivu toteutettuna. Päänavigointi ohjaa ankkurikohtiin ja siirtymä animoitu. Navigointi seuraa mukana selaimen yläreunassa. Ankkurikohdat läpinäkyvien Navigointipalkkien kohdalla. Footeriin liitetty lisätoimintona yhteydenottolomake, joka nousee ylös kosketuksesta. Aktiivinen lomake

*esitetty läpinäkyvänä. Rakenteet ja kuvat mukautuvat näytön tai selaimen leveyteen.  
Toimintojen animoinnit toteutettu jQuery- ja JavaScript-liitännäisillä.*

## **Lähteet**

### **Painetut lähteet**

Fadeyev, D. 2009. Teoksessa Sven Lennarz (toim.) The Smashing Book. User Interface Design in Modern Web Applications. Lybeck: Smashing Media GmbH.

Cattaneo A; Peters Y; Tan, J. 2009. Teoksessa Sven Lennarz (toim.) The Smashing Book. Web typography: Rules, Guidelines and Common Mistakes. Lybeck: Smashing Media GmbH.

Leggett, D; Maier, A. 2009. Teoksessa Sven Lennarz (toim.) The Smashing Book. Usability principles for modern websites. Lybeck: SmashingMedia GmbH.

Kalimo, A. 1996. Graafisen käyttöliittymän suunnittelu - Opas ohjelmistojen käytettävyyteen. Jyväskylä: Gummerus Kirjapaino Oy.

Korpela, J. 2008. CSS verkkosivujen muotoilussa. Porvoo: WS-Bookwell.

Krug, S. 2006. Älä pakota minua ajattelemaan. Jyväskylä: Gummerus Kirjapaino Oy.

Meyer, A. 2006. CSS: the definitive guide. O'Reilly Media, Inc.

Peltomäki, J. 2000. JavaScriptin peruskirja. Tummavuoren kirjapaino Oy.

Saffer, D. 2009. Designing Gestural Interfaces: Touchscreen and Interactive Devices. Sebastopol: O'Reilly Media, Inc.

Saffer, D. 2010. Designing for Interaction: Creating Smart Applications and Clever Devices. Berkeley: AIGA Design Press.

### **Verkkolähteet**

Apple Inc. 2010. Technical Note TN2262: Preparing Your Web Content for iPad - Platform-specific consideration for web content in Safari on iPhone OS devices,

with specific information for iPad. [online] [viitattu 5.3.2011].

<http://developer.apple.com/library/ios/#technotes/tn2010/tn2262/>

Graafinen käyttöliittymä 2010. Wikipedia. [online] [viitattu 5.3.2011].

[http://fi.wikipedia.org/wiki/Graafinen\\_k%C3%A4ytt%C3%B6liittym%C3%A4](http://fi.wikipedia.org/wiki/Graafinen_k%C3%A4ytt%C3%B6liittym%C3%A4)

jQuery. 2010. jQuery. [online] [viitattu 7.4.2011] <http://jquery.com/>

JavaScript Introduction 2011. W3schools. [online] [viitattu 6.3.2011].

[http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp)

Knight, K. 2011. Responsive Web Design: What It Is and How To Use It. Smashing Magazine. [online] [viitattu 15.3.2011].

<http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>

Marcotte, E. 2010a. Fluid Images. UNSTOPPABLE ROBOT NINJA. [online] [viitattu 5.3.2011]. <http://unstoppablerobotninja.com/entry/fluid-images/>

Marcotte, E. 2010. Responsive Web Design. A LIST apart. [online] [viitattu 5.3.2011].

<http://www.alistapart.com/articles/responsive-web-design/>

Martin, M. 2009. Typographic Design Patterns and Best Practises. Smashing Magazine [online] [viitattu 2.12.2010].

<http://www.smashingmagazine.com/2009/08/20/typographic-design-survey-best-practices-from-the-best-blogs/>

Mike. 2010. Mobile Device Detection and Redirection with PHP. Mike's Blog. [online]

[viitattu 5.3.2011]. <http://www.mlynn.org/2010/06/mobile-device-detection-and-redirection-with-php/>

Moline, J. 2010 How to Design a Useful Footer. Smashing Share. [online] [26.1.2010].

viitattu 5.3.2011. <http://smashingshare.com/2011/01/26/how-to-design-a-useful-footer/>

Meyer, A; Meyer, K. 2007. Reset Reasoning. Meyerweb. [online] [viitattu 6.3.2011].  
<http://meyerweb.com/eric/thoughts/2007/04/18/reset-reasoning/>

PHP Introduction 2011. W3schools. [online] [viitattu 6.3.2011].  
[http://www.w3schools.com/php/php\\_intro.asp](http://www.w3schools.com/php/php_intro.asp)

Pilgrim, M. 2010. DIVE INTO HTML5. [online] [viitattu 6.3.2011].  
<http://diveintohtml5.org/>

Päätelaitetunnistus 2009. Wikipedia. [online] [viitattu 6.3.2011].  
<http://fi.wikipedia.org/wiki/P%C3%A4%C3%A4telaitetunnistus>

Rutter, R. 2007. How to Size Text in CSS. A LIST apart. [online] [viitattu 5.3.2011].  
<http://www.alistapart.com/articles/howtosizetextincss/>

Tolvanen, P. 2009. Käsitesekamelskaa: julkaisujärjestelmä, CMS, portaali, sisällönhallintajärjestelmä. Vierityspalkki.fi. [online] [viitattu 6.3.2010].  
<http://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallanhallintajarjestelma/>

What is .htaccess? 2010. .htaccess-Guide. [online] [viitattu 6.5.2011].  
<http://www.htaccess-guide.com/>